AI 如何学会写代码？

从自动补全到自主代理——AI 编程工具进化全史 × OpenAI Codex 深度指南

AI 编程工具经历了三次革命性跃迁

每一代都将开发者从重复劳动中解放出来，推向更高层次的创造力

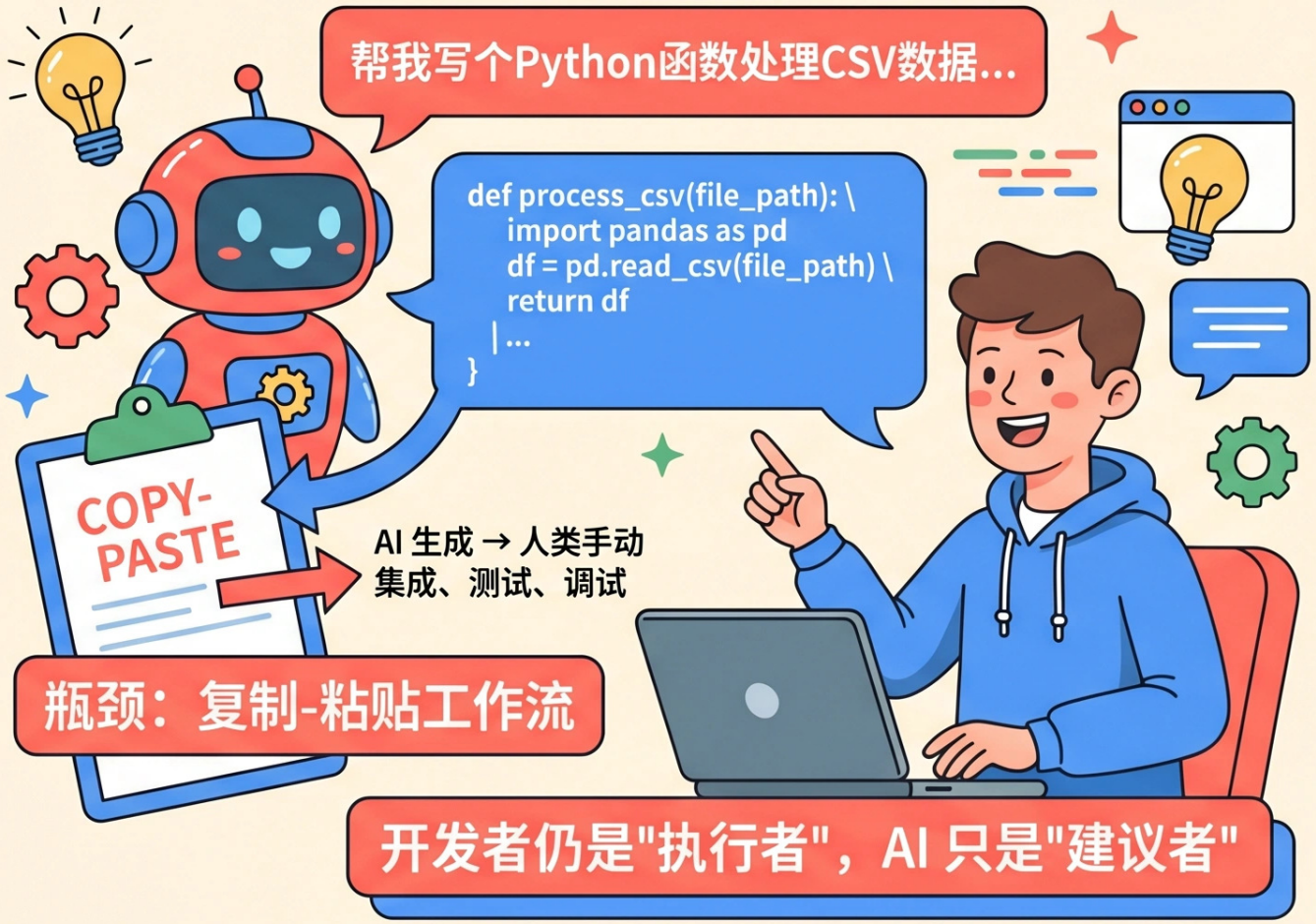
GitHub Copilot 开创了"幽灵文本"时代

- 2021年发布，基于 OpenAI Codex 模型
- 核心能力：预测并建议下一行代码（代码自动补全进化版）
- 响应式工具：开发者输入，AI 建议，人类决策
- 意义：首次证明 AI 可理解代码语义
- 局限：无法跨文件理解、不能执行代码、不能自我纠错

ChatGPT 让每个人都能"聊出"代码

AI 编程工具进化全史 × OpenAI Codex 深度指南

- 2022-2024年，ChatGPT 与 Claude 普及对话式编程
- 开发者可通过自然语言描述需求，获得完整函数、类或文件
- 突破：不再需要记忆 API 语法，降低编程门槛
- 瓶颈：复制-粘贴工作流 —— AI 生成，人类手动集成、测试、调试
- 开发者仍是"执行者"，AI 只是"建议者"



2025年：AI 代理颠覆了"谁在写代码"这个问题

- 代理型 AI：自主规划、跨文件执行、运行终端命令、自我纠错

- 氛围编程（Vibe Coding）：开发者描述高层目标，AI 负责实现、测试、部署

- 角色转型：从'编写代码逻辑'转向'治理代理系统'

- 核心技能：架构设计、需求定义、上下文管理

编程史上最深刻的范式转移

OpenAI Codex 2026：专为异步委派而生的代理模型

OpenAI Codex

- **Codex 已演进为 GPT-5 家族专业代理模型** (gpt-5.3-codex)
- **核心设计哲学：异步委派 + 高智力推理**
- **三大访问方式：**
  - **Codex CLI** — 终端自动化，DevOps 脚本，专业开发者首选
  - **Codex Mac 桌面应用** — 可视化 Diff，多线程任务管理
  - **IDE 插件** — VS Code 无缝集成，上下文感知重构
- **Token 消耗效率远高于竞品，适合长周期自主任务**

一次运行即成功的提示词策略

成功标准

策略一：定义详尽的成功标准

— 预加载所有技术要求（库选择、JWT 过期时间、验证逻辑），避免模糊指令

环境约束

策略二：明确环境与数据约束

— 告知 API 分页方式、数据格式，防止 Codex 基于训练假设产生偏差

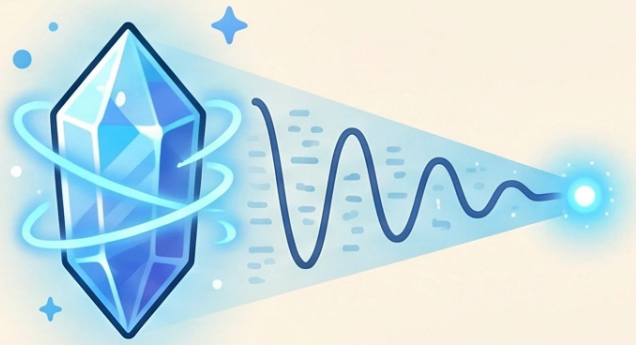
验证循环

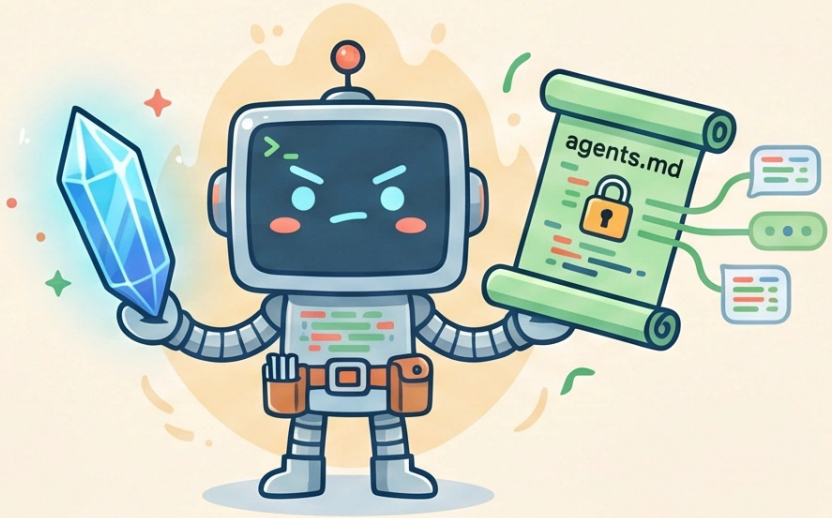
策略三：建立验证循环

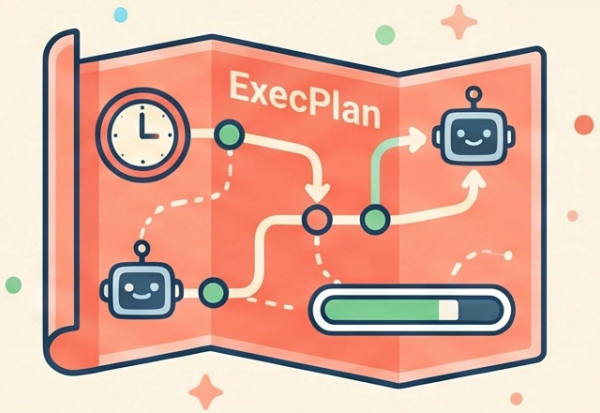
— 要求模型编写测试用例或运行 Linter，利用 AI 自我验证能力

核心原则：上下文越丰富，结果越精准

三大高级功能让 Codex 跨越时间与复杂度







上下文压缩（Compaction）

处理跨越数小时的长任务，不
不因 Token 限制丢失初始目标

持久化指令（agents.md）

存储编码规范和项目目标，
确保所有会话一致性

执行计划（ExecPlans）

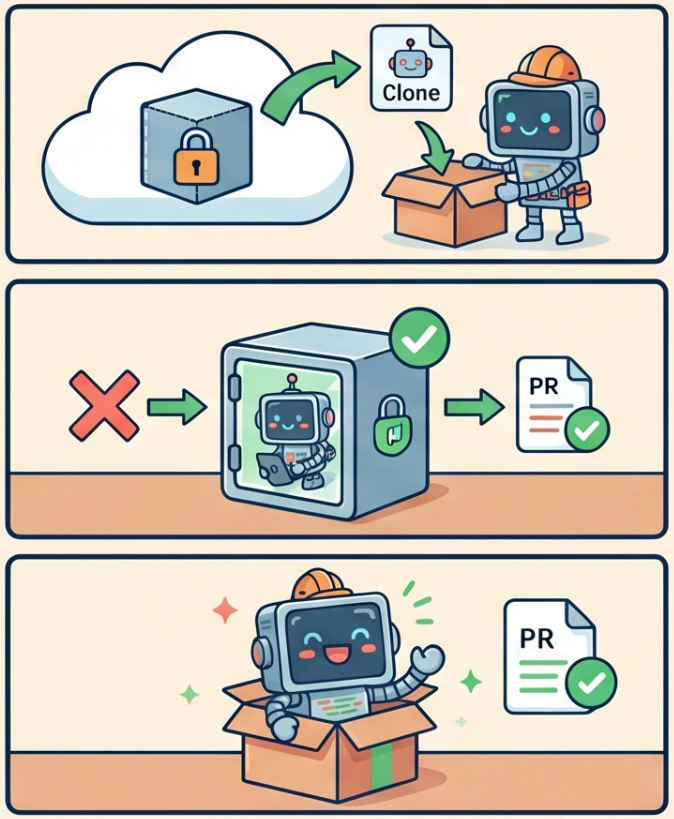
记录决策历史和进度，支持
协同跟踪长周期任务

三者组合：实现真正意义上的跨会话、跨时间的持续自主工作

三种专业工作流模式，解锁 Codex 全部潜力

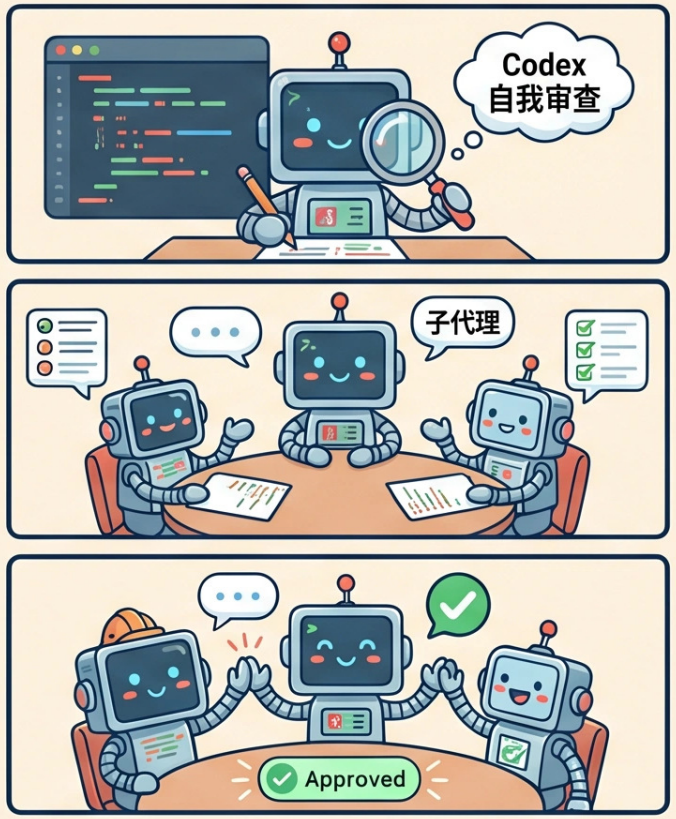
核心思想：让 AI 代理互相监督，人类只需定义终态

1. 云端沙盒执行



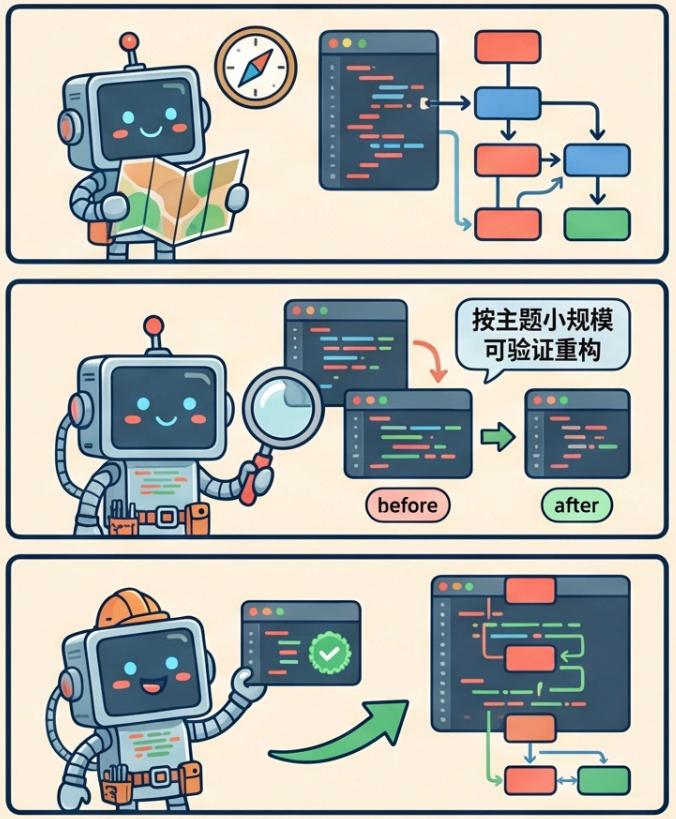
- 隔离云端线程中克隆仓库执行任务，安全处理不受信任的 PR

2. 代理审查循环



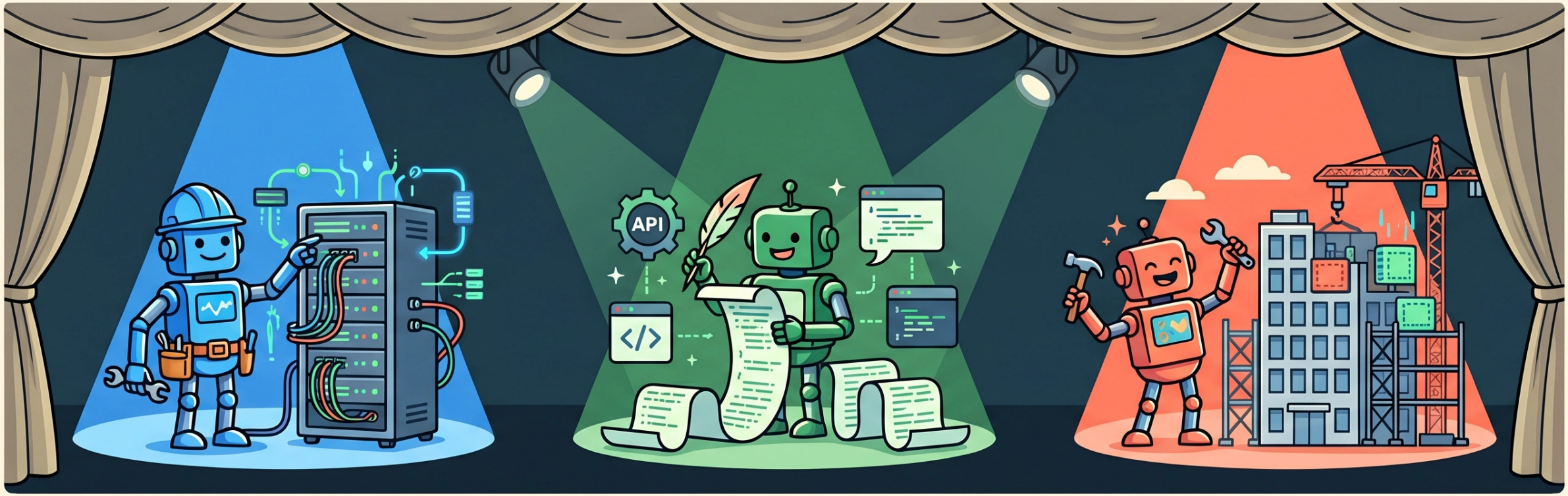
- Codex 自我审查 → 子代理评审 → 达成一致后提交

3. 小步快跑式重构



- 映射代码库 → 按主题小规模可验证重构

Codex 在这三个场景中表现无可匹敌



DevOps 自动化：CI/CD 流水线搭建、基础设施即代码、监控脚本
— Codex 可独立完成端到端

脚本编写：数据处理、API 集成、批量文件操作
— 一次描述，自动生成并验证

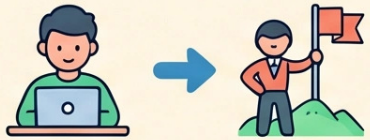
大规模自主重构：跨越数百文件的代码库现代化改造
— ExecPlans + 小步重构组合拳

成本优势：Token 消耗效率远高于竞争工具，长周期任务首选

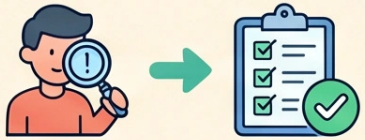
建议：将 Codex 定位为'自主执行引擎'，而非'代码建议工具'

未来的开发者：代理系统的架构师与治理者

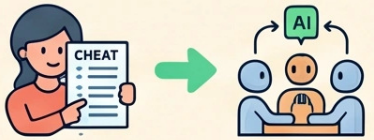
- **AI 编程工具**的终极影响不是“取代开发者”，而是**重新定义开发者的价值**



从“编写代码逻辑”→“定义系统目标与约束”



从“调试 Bug”→“设计验证与审查机制”



从“记忆 API 语法”→“管理上下文与代理协作”

最重要的新技能：清晰表达意图、设计可验证的成功标准、治理多代理系统

氛围编程不是懒惰，而是更高层次的工程思维

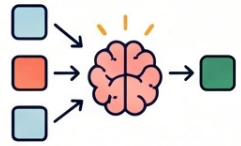
三个时代，一个趋势：AI 正在成为你的编程搭档

历史脉络

自动补全 → 对话助手
→ 代理型 AI，
每一步都是质的飞跃

Codex 核心

异步委派 +
高智力推理 +
持久化上下文 =



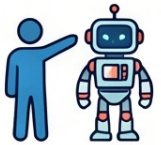
真正的自主编程代理

使用精髓

丰富上下文 +
验证循环 +
小步迭代
= 一次运行即成功

未来方向 & 行动建议

- 开发者成为代理治理者，
  AI 负责执行，
  人类负责方向



- 立即尝试 Codex CLI，
  从一个真实的 DevOps
  脚本任务开始